# Cone Schedules for Processing Systems in Fluctuating Environments

KEVIN ROSS [1]
NICHOLAS BAMBOS [2]
GEORGE MICHAILIDIS [3]

**Abstract**

We consider a generalized processing system having several queues, where the available service rate combinations are fluctuating over time due to reliability and availability variations. The objective is to allocate the available resources, and corresponding service rates, in response to both workload and service capacity considerations, in order to maintain the long term stability of the system. The service configurations are completely arbitrary, including negative service rates which represent forwarding and service-induced cross traffic. We employ a trace-based trajectory asymptotic technique, which requires minimal assumptions about the arrival dynamics of the system.

We prove that *cone schedules*, which leverage the geometry of the queueing dynamics, maximize the system throughput for a broad class of processing systems, even under adversarial arrival processes. We study the impact of fluctuating service availability, where resources are available only some of the time, and the schedule must dynamically respond to the changing available service rates, establishing both the capacity of such systems and the class of schedules which will stabilize the system at full capacity. The rich geometry of the system dynamics leads to important insights for stability, performance and scalability, and substantially generalizes previous findings.

The processing system studied here models a broad variety of computer, communication and service networks, including varying channel conditions and cross-traffic in wireless networking, and call centers with fluctuating capacity. The findings have implications for bandwidth and processor allocation in communication networks and workforce scheduling in congested call centers. By establishing a broad class of stabilizing schedules under general conditions, we find that a scheduler can select the schedule from within this class that best meets their load balancing and scalability requirements.

**Keywords:** random environment, stability, adversarial queueing theory, dynamic scheduling, throughput maximization.

## 1 Introduction

We consider a processing system comprised of $Q$ infinite capacity queues, indexed by $q \in \mathcal{Q} = \{1, 2, ..., Q\}$, operating in a time-varying environment which fluctuates amongst environment states $e \in \mathcal{E} = \{1, 2, ..., E\}$. In each environment state, only a subset of the service configurations are available. The process scheduler selects a service configuration vector $S = (S_1, \cdots, S_Q)$ from the environment-dependent available set $\mathcal{S}^e$. Upon selection, if $S_q > 0$ then queue $q$ is emptied at rate $S_q$, and if $S_q < 0$ then the queue is filled at the corresponding rate. The available service configurations can be completely arbitrary, including vectors with any combination of positive and negative components.

A key question addressed in this study is which of the available service configurations should be selected, given the system workload and environment state histories, so as to maximize its throughput. We

[1]School of Engineering, University of California Santa Cruz; kross@soe.ucsc.edu;
[2]Electrical Engineering and Management Science & Engineering, Stanford University; bambos@stanford.edu
[3]Statistics and Electrical Engineering & Computer Science, The University of Michigan; gmichail@umich.edu

introduce a family of resource allocation policies - called Cone Schedules - which are shown to stabilize the system under the maximal possible traffic load, even if that load is designed by an adversary to destabilize the system whenever possible.

This canonical processing model captures several applications in computing and communication systems, including wireless networks, packet switches and call centers. The main characteristic of these applications is that the service rates across multiple queues are coupled through operational constraints, giving rise to the available service configurations. Service rate availability (corresponding to the environment states) is affected by staff scheduling in call centers, congestion dynamics in wireless networks and scheduled or unscheduled outages due to maintenance or reliability issues in other processing systems.

### 1.1 Related Work

The trace-based stability analysis technique employed in this paper relates to the study of adversarial queueing networks exemplified in [Andrews et al., 2001] and [Borodin et al., 2001]. This approach avoids imposing a probabilistic framework on the arrival traffic, and instead analyzes the performance of a queueing network under and adversarial arrival traffic trace, designed to stress the system as much as possible. They describe a queueing network as universally stable when they can show that the total workload of the system is bounded under any deterministic or stochastic adversary's arrival trace. This work is really finding the *worst-case* behavior of a network by considering the network to be a *game* between the schedule (protocol) and the worst possible arrival trace (adversary). They limit the absolute arrival volume within a finite interval, but do not require it to follow any stationary distribution or apply any further restrictions. This concept builds upon earlier work called *leaky-bucket* analysis in [Cruz, 1991a] and [Cruz, 1991b].

Adversarial models have been used to in packet networks before, such as [Borodin et al., 2001] which considers a fixed-path packet network. Some more general queueing systems, including multiclass queueing networks are studied in [Tsaparas, 2000], with generalized service times and heterogeneous customers. Adversarial methods have also been employed to study multi-hop network stability in [Kushner, 2006]. In [Anshelevich et al., 2002], adversarial models are used to analyze load-balancing algorithms in a distributed setting based using a token-based system on a network with limited deviations from the average load. While none of these study the same network scheduling setting of this paper (to our knowledge they have only considered fixed-path networks under time-invariant service environments), each example presents a persuasive argument for the value of network stability analysis in the absence of a well-defined probabilistic framework.

A special example of the system described in this paper is a single crossbar packet/cell switch with virtual output queues, used in high speed IP networks. The switch paradigm is the focus of [Ross and Bambos, 2009], and provides a helpful context to develop the cone algorithms. In this switch, cells arriving to each input port get buffered in separate virtual queues, based on the output port they are destined to. The switching fabric can be set to a different connectivity mode in each time slot, matching each input port with a corresponding output port for cell transfer. In this context, *Maximum Weight Matching* (MWM) has been shown in [McKeown et al., 1999] to maximize the throughput of input queued switches, employing Lyapunov methods for stability analysis, as also in constrained queueing systems studied in [Tassiulas and Ephremides, 1992, Tassiulas, 1995, Tassiulas and Bhattacharya, 2000, Hung and Michailidis, 2011]. In our more general service model, MWM corresponds to maximizing $\langle S, X\rangle = \sum_q S_q X_q$, where the weight $X_q$ is the cell workload of queue $q$ or a related congestion measure, and the $S$ vectors represent the crossbar configurations.

More general results on the stability of MWM algorithms, using fluid scaling methods, were later

obtained in [Dai and Prabhakar, 2000], and on a generalized switch model in [Stolyar, 2004]. Stability in networks of switches was studied in [Marsan et al., 2005] and [Leonardi et al., 2005]. [Dai and Lin, 2005] and [Dai and Lin, 2008] considered maximum pressure policies by modeling fluid flows for types of processing networks. Their work can be seen as a generalization of the policies which maximize $\sum_q S_q X_q$ where some of the service rates are negative because the available configurations involve forwarding workload from one queue to another downstream queue. [Neely et al., 2003] studied broader optimal controls for generalized (wireless) network models that involve joint scheduling, routing and power allocation. All of these have significantly advanced the theory of the stability of scheduling rules which allocate service to queues based on a weighted-matching approach, and utilize a probabilistic framework to apply fluid limit or heavy-traffic analysis.

Instead of using fluid scaling methods (primarily analytic, involving passage to a limit regime) to establish the results, we opt to use an alternative direct and primarily geometric approach in this work, which seems to have broader applicability to other queueing systems and reveals useful geometric insight regarding their dynamics. The trace-based asymptotic analysis employed here was introduced in [Armony and Bambos, 2003], where the maximum weight matching algorithms were studied and it was shown that maintain maximal throughput is guaranteed under very general arrival process assumptions. The method was also employed in [Bambos and Michailidis, 2004] where randomly fluctuating service levels were studied. In that case the service rate assignments are made without full knowledge of service availability, as opposed to the processing systems studied here where service allocation decisions are made in response to availability. Like the adversarial queueing models, there is no probabilistic framework required, but unlike the traditional adversarial models, there is also no short-term restriction on arrival bursts in finite time, but just a long-term traffic load restriction. This leads to more general stability results, but eliminates the possibility of tighter bounds on other performance metrics. For example under such general assumptions there can be no guaranteed finite bound on the total workload, or even the expected workload in the system.

### 1.2 Results Overview

We classify the stability region for these processing systems with fluctuating service availability. We find that rate stability for these general processing systems can be guaranteed by the class of *cone schedules*, for any arbitrary arrival process that can possibly be stabilized. Cone schedules use the available service vector with maximal projection $\langle S, \mathbf{B}X \rangle = \sum_p \sum_q S_p B_{pq} X_q$ on the *projected* workload vector $\mathbf{B}X$, for every matrix that is *positive-definite*, has *negative or zero off-diagonal* elements, and is *symmetric*. This substantially generalizes a similar result in [Ross and Bambos, 2009] , where the same class of algorithms was shown to maximize throughput for the special case of packet switches.

In classifying the stability region, we show how the combination of service vectors in each environment impacts the overall capacity of the system, beyond the long term availability of each service vector. The geometric framework for stability aids the intuition and analysis significantly. Because of environment fluctuations, one may expect that a scheduling rule needs to account for future and past states. However we find that cone schedules, which respond only to the current workload, are able to guarantee stability for any arrival rate within the stability region.

The service rates in this paper are allowed to be completely arbitrary, in contrast to previous results using the trace-based analysis which only applied to positive-service switches. This captures cross-traffic and forwarding between queues, because the selected service vector may induce additional workload to the system, in addition to the external arrival process. Further, in this work time is continuous, and arbitrarily large arrival bursts can be handled at arbitrarily small time intervals. This is more general than previous

models where arrivals and decisions were restricted to timeslots.

From an architectural point of view, the geometric approach to the scheduling problem provides key practical design leads. Specifically, the conic representation (Section 4) of cone schedules leads to scalable implementations in switching systems. Further, varying the elements of matrix $\mathbf{B}$, we can generate a very rich family of cone schedules that implement a soft *coupled priority scheme* (and coupled load balancing) across the various queues, managing delay tradeoffs between them. The schedules are also robust to any sublinear perturbation such as delayed or flawed state information.

The remainder of the paper proceeds as follows. In Section 2, we introduce the model and system dynamics. Section 3 describes the throughput capacity or stability region of these networks, and in section 4 we introduce the family of Cone Schedules and their geometry. Stability and performance implications are discussed in sections 5 and 6 respectively. We conclude in Section 7.

## 2 The Processing Structure

Let $\int_0^t A_q(z)dz$ be the total workload that arrives to queue $q$ in the time interval $(0,t]$; that is, $A_q(t) \geq 0$ is the instantaneous workload arrival rate at time $t \geq 0$. The traffic trace $\mathbf{A}_q = \{A_q(t), t \geq 0\}$ is a (deterministic) function, which may have *discontinuities* and even $\delta$*-jumps* for each $q \in \mathcal{Q}$. The overall (vector) instantaneous traffic rate is $A(t) = (A_1(t), A_2(t), ..., A_q(t), ..., A_Q(t))$ at time $t > 0$ and the *traffic trace* is $\mathbf{A} = \{A(t), t \geq 0\}$. We assume that the (long-term) *traffic load* of the trace[4] $\mathbf{A}$,

$$\lim_{t\to\infty} \frac{\int_0^t A(z)dz}{t} = \rho(\mathbf{A}) \in \mathbb{R}_{0+}^Q, \tag{2.1}$$

is well-defined on the traffic trace $\mathbf{A}$. Correspondingly, we define the set of traffic traces of load $\rho \in \mathbb{R}_{0+}^Q$,

$$\mathfrak{A}(\rho) = \left\{ \mathbf{A} = \{A(t), t \geq 0\} : \lim_{t\to\infty} \frac{\int_0^t A(z)dz}{t} = \rho \right\}, \tag{2.2}$$

restricting our attention in this paper to traffic traces of well-defined load. A variety of natural arrival processes are included. For example, $A_q(t) = \sum_{j=1}^{\infty} \sigma_q^j \, \mathbf{1}_{\{t=t_q^j\}}$ models jobs of service requirement $\sigma_q^j$ arriving at times $t_q^j > 0$ to queue $q$. In this case, $A_q(t)$ is zero between consecutive $\delta$-jumps. In general, there could be *positive* instantaneous workload arrival rate between consecutive $\delta$-jumps, which would represent a continuous inflow of work.

No further restrictions are placed on the arriving traffic trace. It may be generated by an underlying stochastic process, or even an adversary specifically designed to destabilize the system whenever possible.

The arriving workload is queued up in the queues $q \in \mathcal{Q}$, which are assumed to be of infinite capacity. Let $X_q(t)$ be the workload (total workload or service requirement) in queue $q$ at time $t \geq 0$ and

$$X(t) = (X_1(t), X_2(t), ..., X_q(t), ..., X_Q(t)) \in \mathbb{R}_{0+}^Q,$$

the overall (vector) workload.

[4]Throughout this study we employ the notation $\mathbb{Z}_+ = \{1, 2, 3, \cdots\}$, $\mathbb{Z}_{0+} = \{0, 1, 2, \cdots\}$, $\mathbb{R} = (-\infty, \infty)$, $\mathbb{R}_+ = (0, \infty)$, $\mathbb{R}_{0+} = [0, \infty)$

The processing system operates in a *fluctuating environment*, which can be in one of $E$ distinct states at any point in time, indexed by $e \in \mathcal{E} = \{1, 2, \cdots, E\}$. Let $e(t) \in \mathcal{E}$ be the environment state at time $t$ and $\mathbf{E} = \{e(t),\ t \in \mathbb{R}\}$ the overall environment trace over time. It is assumed that the proportion of time the environment trace $\mathbf{E}$ spends in each state $e \in \mathcal{E}$ is well-defined, that is,

$$\lim_{t\to\infty} \frac{\int_0^t \mathbf{1}_{\{e(z)=z\}} dz}{t} = \pi^e(\mathbf{E})$$

with $\sum_{e\in\mathcal{E}} \pi^e(\mathbf{E}) = 1, \pi^e(\mathbf{E}) > 0, e \in \mathcal{E}$. Correspondingly, we define the set of environment traces $\mathbf{E}$ with time proportions $\pi^e, e \in \mathcal{E}$ as

$$\mathfrak{E}(\pi^e, e \in \mathcal{E}) = \left\{ \mathbf{E} = \{e(t), t \geq 0\} : \lim_{t\to\infty} \frac{\int_0^t \mathbf{1}_{\{e(z)=z\}} dz}{t} = \pi^e, e \in \mathcal{E} \right\}, \tag{2.3}$$

and restrict our attention in this paper to environment traces that have well-defined time proportions. Finally, $E = 1$ naturally corresponds to the degenerate case of a constant (non-fluctuating) environment.

When the environment is in state $e \in \mathcal{E}$, a (nonempty) set of *service vectors* $\mathcal{S}^e$ becomes available to the system manager, who can select a service vector $S \in \mathcal{S}^e$ at any point in time to operate the system. Each $S \in \mathcal{S}^e$ is a $Q$-dimensional vector

$$S = (S_1, S_2, ..., S_q, ..., S_Q) \in \mathbb{R}^Q,$$

where $S_q \in \mathbb{R}$ is the drain (or fill, see below) rate of queue $q$ when the service vector $S$ is used. For example, in a simple system with two queues ($Q = 2$), a service vector $S^1 = (1.35, 2.17)$ would serve (drain) queue 1 at rate 1.35 and queue 2 at rate 2.17 (work units per time unit). This is the standard way of viewing service vectors.

In this general model, however, we also allow for *negative* 'service' rates, actually corresponding to traffic workload 'feed' rates, as explained below. In the previous simple example of two queues, a service vector $S^2 = (1.2, -0.8)$ would serve (drain) the first queue at rate 1.2, but feed workload to the second queue at rate 0.8, filling it up.

The motivation to allow for negative components $S_q < 0$ in the service vectors $S \in \mathcal{S}^e$ comes from the need to model *environmental (background) cross-traffic* sharing the queue buffers with the primary (foreground) traffic $\{A(t), t \geq 0\}$. This cross-traffic depends explicitly on the service vector $S \in \mathcal{S}^e$ used, and implicitly on the environment state $e \in \mathcal{E}$ through the set $\mathcal{S}^e$ where the service vector $S$ should be chosen from. When service vector $S$ is used with $S_q < 0$ for some queue $q \in \mathcal{Q}$, this corresponds to cross-traffic workload fed into queue $q$ at constant rate $-S_q > 0$, *in addition* to the primary traffic workload $\{A_q(t), t \geq 0\}$. It is easy to see that $-S_q > 0$ can be interpreted as the 'net' cross-traffic through the queue; that is, workload could be fed into queue $q$ at rate $r_1 > 0$ and removed (served) at rate $r_2 > 0$, with the net cross-traffic load fed into the queue being $-S_q = r_1 - r_2$.

One special case related to this model is a feed-forward network. A service vector representing the transfer of workload from one upstream queue $q^u$ to another downstream queue $q^d$ would be represented with $S_{q^u} = -S_{q^d}$ and all other $S_q = 0$. The model here could handle the aggregate of many transfers, as well as gain and loss in the system at any queue. The concept of cross-traffic considered here is more general, requiring no restrictions on the physical structure of the network. Feed-forward networks require some additional assumptions and are not the primary focus of this paper, but have been studied extensively elsewhere, such as [Dai and Lin, 2005].

Note that the above environmental cross-traffic is far less 'innocuous' than simply allowing the primary traffic to be modulated[5] by the environment state. Indeed, the cross-traffic depends on the choice of service vector $S$, hence, the scheduling decisions actively influence it. The environment plays only a secondary role by defining $\mathcal{S}^e$, hence, restricting the range of scheduling choices. Actually, the introduction of cross-traffic is shown to have significant implications on the stability behavior of the scheduling policies studied later.

The sets $\mathcal{S}^e, e \in \mathcal{E}$ may be overlapping, that is, a service vector may be available under one or more environment states. Let $\mathcal{S} = \bigcup_{e\in\mathcal{E}} \mathcal{S}^e$. It is assumed that each service vector set $\mathcal{S}^e, e \in \mathcal{E}$ is *complete*, that is, for each $e \in \mathcal{E}$ and any $q \in \mathcal{Q}$

$$(S_1, S_2, ..., S_{q-1}, S_q > 0, S_{q+1}, ..., S_Q) \in \mathcal{S}^e \Rightarrow (S_1, S_2, ..., S_{q-1}, S_q = 0, S_{q+1}, ..., S_Q) \in \mathcal{S}^e. \quad (2.4)$$

Hence, any 'sub-vector' of a service vector in $\mathcal{S}^e$ (i.e. with one or more *positive* components reduced to zero) is also[6] a service vector in $\mathcal{S}^e$. The reason for requiring *completeness* of each $\mathcal{S}^e$ is to accommodate the following situation: *when some queues become empty and ceases receiving service, the resulting effective service vector is a feasible one.* Under the latter perspective, the imposed assumption (2.4) is a natural one indeed. As seen below, it allows us to naturally handle schedules which provide zero service rate to empty queues.

The key issue is choosing the service vector $S(t) \in \mathcal{S}^{e(t)}$ at time $t$, when the environment is in state $e(t)$ and the vectors $\mathcal{S}^{e(t)}$ are available to choose from. In general, the decision can be based on the observable histories of the workload $\{X(z), z \leq t\}$, the environment $\{e(z), z \leq t\}$, and prior service choices $\{S(z), z < t\}$. The scheduling policy is the overall trace of service vector choices $\mathbf{S} = \{S(t), t \geq 0\}$. Our primary objective is to design schedules $\mathbf{S}$ which maximize the system throughput (keep the system stable under the maximum possible load $\rho$), while being *robust* and utilizing minimum information, like the current workload and environment states, with no knowledge of the actual load $\rho$ and the environment time proportions $\{\pi^e, e \in \mathcal{E}\}$. We elaborate on such issues later.

We are interested in natural schedules $\mathbf{S} = \{S(t), t \geq 0\}$ that never apply positive service to empty queues. That is, whenever $X_q(t) = 0$ the scheduler chooses a service vector $S(t) \in \mathcal{S}^{e(t)}$ with $S_q(t) \leq 0$. This is possible because we have assumed that the sets $\mathcal{S}^e, e \in \mathcal{E}$ are *complete*. Therefore, we can write

$$X(t) = X(0) + \int_0^t A(z)dz - \int_0^t S(z)dz \quad (2.5)$$

without having to explicitly 'compensate' for any idling time.

[5]Actually, the environment could also modulate the primary traffic trace $\{A(t), t \geq 0\}$ in the following sense. There is a collection of traffic traces $\{A^e(t), t \geq 0\}$ one for each environment state $e \in \mathcal{E}$. When the environment is in state $e$, the traffic driven into the system is selected from $\{A^e(t), t \geq 0\}$. Therefore, the overall traffic trace is simply $\{A(t) = \sum_{e\in\mathcal{E}} A^e(t)\mathbf{1}_{\{e(t)=e\}}, t \geq 0\}$. Hence, this basically reverts to the standard model (as long as the limit $\lim_{t\to\infty} \int_0^t A(z)dz/t$ exists) and this is why we do not treat this case explicitly.

[6]Note that if any service vector in $\mathcal{S}^e$ has no negative components, then the zero vector $(0, 0, ..., 0)$ must be in $\mathcal{S}^e$ as a sub-vector of the former vector, due to completeness. But if each service vector in $S^e$ has at least one negative component, the zero vector does not necessarily have to be in $\mathcal{S}^e$ unless it is by design.

# 3 The Stability Issue

In the interest of robustness of the results, we employ the 'lightest' possible (see below) concept of stability, that is, *rate stability* [Bambos and Walrand, 1993]. Specifically, we call the system stable iff

$$\lim_{t\to\infty}\frac{X(t)}{t}=\lim_{t\to\infty}\left(\frac{X_1(t)}{t},\frac{X_2(t)}{t},...,\frac{X_q(t)}{t},...,\frac{X_Q(t)}{t}\right)=0. \tag{3.1}$$

Note that from (2.5) and (2.1), rate-stability implies that $\rho=\lim_{t\to\infty}\{\int_0^t S(z)dz/t\}$. Moreover, when the traffic trace involves pure 'job-arrivals' ($\delta$-jumps) with zero workload arrival rate between them, then rate-stability (3.1) implies that the long-term job departure rate from each queue is equal to the long-term job arrival rate [Armony and Bambos, 2003]. Therefore, there is *flow conservation* through the system and the inflow at each queue is equal to the outflow. On the contrary, when the system is unstable there is a inflow-to-outflow deficit, which accumulates in the queues. This is consistent with engineering intuition and, in that sense, the concept of rate-stability is quite natural. Of course, it can be further tightened by imposing progressively heavier statistical assumptions on the traffic and environment traces. We resist doing that at this point, in order to preserve the generality of the results and keep them as robust and 'assumptions-agnostic' as possible.

**Definition 3.1 (Stability Region)** We define formally the stability region $\mathcal{R}$ of the system as the set of traffic loads $\rho\in\mathbb{R}^Q_{0+}$ for which there exists a scheduling policy $\mathbf{S}=\{S(t),t\geq 0\}$ under which the system is rate-stable (3.1) for *all* traffic traces $\mathbf{A}=\{A(t),t\geq 0\}$ with $\rho(\mathbf{A})=\rho$ and *all* environment traces $\mathbf{E}=\{e(t),t\geq 0\}$ with $\pi^e(\mathbf{E})=\pi^e, e\in\mathcal{E}$.

As shown below, the universal stability region $\mathcal{R}$ can be characterized as

$$\mathcal{R}(\mathcal{S}^e,\pi^e,e\in\mathcal{E})= \left\{\rho\in\mathbb{R}^Q_+ : 0\leq\rho\leq\sum_{e\in\mathcal{E}}\pi^e\sum_{S\in\mathcal{S}^e}\phi^e_S S,\ \text{for some } \phi^e_S\geq 0 \text{ with } \sum_{S\in\mathcal{S}^e}\phi^e_S=1,\ e\in\mathcal{E}\right\} \tag{3.2}$$

The intuition is that $\rho$ is in the stability region $\mathcal{R}$ if it is dominated (covered) by a convex combination of the service vectors $S\in\mathcal{S}$, induced under the various service vectors in $\mathcal{S}^e, e\in\mathcal{E}$. Thus, $\mathcal{R}$ is the 'weighted sum' of the various 'stability regions' generated by the individual sets $\mathcal{S}^e$ for each state $e\in\mathcal{E}$ of the environment.

If $\rho\in\mathcal{R}$ and $\pi^e, e\in\mathcal{E}$ were known in advance and $\phi^e_S$ could be computed, then selecting each mode $S\in\mathcal{S}^e$ for a fraction $\phi^e_S$ of the time while the system is in environment state $e\in\mathcal{E}$ would keep the system stable. This could be achieved through round-robin or randomized algorithms. A scheduling algorithm which maintains stability (3.1) for any $\rho\in\mathcal{R}$ is referred to as *throughput maximizing*. However, we are primarily interested in adaptive scheduling schemes which maintain stability (3.2) for all $\rho\in\mathcal{R}$, without actual prior knowledge of $\rho$ or $\pi^e$. The *cone schedules* defined below are shown to provide such universal stability for any traffic load in $\mathcal{R}$, while being agnostic to particulars of the traffic and environment traces $\rho(\mathbf{A})$ and $\pi^e(\mathbf{E})=\pi^e, e\in\mathcal{E}$; they respond only to current workload and environment state.

In general, the stability behavior of scheduling rules could require the arrival trace to satisfy stronger conditions than those above. For example, restricting the study to Markovian or stationary arrival processes,

or disallowing mixing, may provide special cases of stability. Instead, we allow the arrival traffic trace $\mathbf{A}$ and environment trace $\mathbf{E}$ to be designed by an adversary to stress the system. Consider for example an arrival trace where arrivals to queue $q$ are deliberately correlated to the environment states when $q$ cannot be served at maximum capacity. Even further, an adversarial trace may push arrivals to queues in a state-dependent way which responds to the scheduling rules themselves. These are very difficult to capture by a natural probabilistic framework, but are simply treated as particular traffic traces here.

To motivate the definition of the stability region for the processing system under consideration, we examine first the case where $S_q \geq 0$ for all $q \in \mathcal{Q}$ and there is only one environment state ($E = 1$, no environment fluctuation); that is, service is always non-negative and all service vectors $\mathcal{S}$ are available at every point in time. Under the trace-based perspective employed in this paper, it is known [Armony and Bambos, 2003] that for any load $\rho$ in the region

$$\left\{ \rho \in \mathbb{R}_{0+}^{Q} : \ \rho \leq \sum_{S \in \mathcal{S}} \phi_S S \text{ for some } \phi_S \geq 0, \ S \in \mathcal{S}, \ \text{ such that } \sum_{S \in \mathcal{S}} \phi^S \leq 1 \right\}$$

the system can be made rate stable with an appropriate scheduling rule. The non-negative parameters $\phi_S, S \in \mathcal{S}$ are essentially proportional weights, which are chosen so that the load vector $\rho$ is component-wise dominated by the weighted linear combination $\sum_{S \in \mathcal{S}} \phi_S S$ of the service vectors.

Extending this 'geometric' stability perspective to allow cross traffic and varying environment states is not a trivial task. Intuition may suggest that the stability region in networks in fluctuating environments should be reduced according to how often each mode is available. Consider the following simple network to illustrate that the distribution of environment states $\{\pi^e, e \in \mathcal{E}\}$ is critical to stability. Take a 2-queue network with three service vectors, $S^1 = (1, 0), S^2 = (0, 1), S^3 = (1, 1)$. Clearly, if all vectors are available all the time, by employing always $S^3$ the system can accommodate any input vector $(\rho_1, \rho_2) \in [0, 1]^2$. On the other hand, if there are two environment states $\mathcal{E} = \{e_1, e_2\}$ with service vector sets $\mathcal{S}^{e_1} = \{S^1, S^2\}$ and $\mathcal{S}^{e_2} = \{S^3\}$ with $\pi^{e_1} = 0.5, \ \pi^{e_2} = 0.5$, then the system can accommodate any input vector $\rho \geq 0$ satisfying the conditions $\rho_1 + \rho_2 \leq 1.5, \rho_1 \leq 1$, and $\rho_2 \leq 1$.

However, a different configuration of the service vector sets, say $\mathcal{S}^{e_1} = \{S^1\}$ and $\mathcal{S}^{e_2} = \{S^2, \ S^3\}$ with $\pi^{e_1} = 0.5, \pi^{e_2} = 0.5$, yields $\rho_1 \in [0, 1]$ and $\rho_2 \in [0, 0.5]$ for stability. Note that although the sets $\mathcal{S}^{e_1}$ and $\mathcal{S}^{e_2}$ ensure that each service vector is available for the same portion of time in both scenarios, the relative combinations of the available service vectors change the stability region. We illustrate (and generalize) this perspective in Figure 1.

We establish first that if $\rho \notin \mathcal{R}$, it is impossible to maintain stability and flow conservation in all queues, no matter what scheduling policy one employs. At least one queue will suffer an outflow deficit (compared to its inflow), which will accumulate in the queue and cause its workload to explode linearly it in time.

**Proposition 3.1 (Instability)** For any arbitrarily fixed traffic traffic trace $\mathbf{A}$ and environment trace $\mathbf{E}$, we have

$$\rho(\mathbf{A}) \notin \mathcal{R} \Longrightarrow \limsup_{t \to \infty} \frac{X_q(t)}{t} > 0, \tag{3.3}$$

for at least one queue $q \in \mathcal{Q}$ under *any* scheduling policy.

**Proof:** For convenience, we drop the fixed argument $\mathbf{A}$ from $\rho(\mathbf{A})$ and write it traffic load as simply $\rho$,

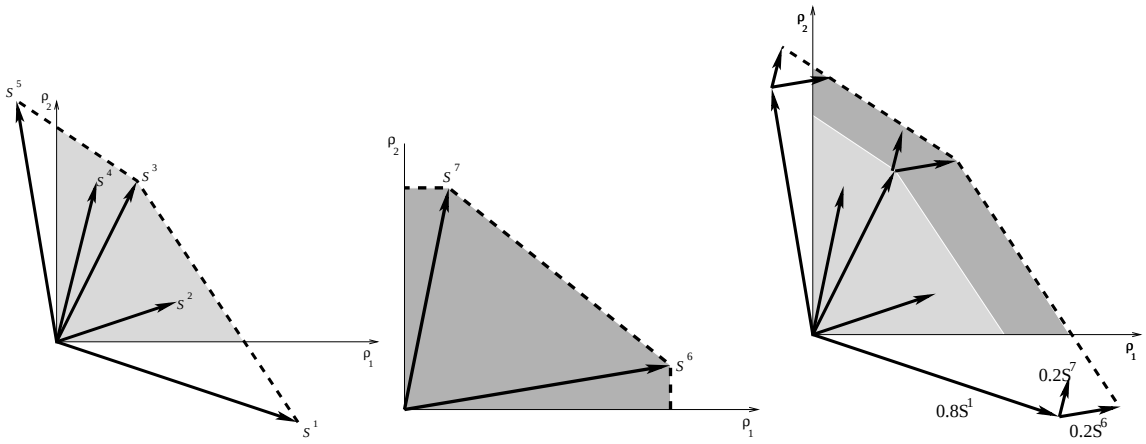



Figure 1: **The stability region.** The set of allowable arrival rate vectors $\rho$ is called the stability region $\mathcal{R}$. Two separate sets of service vectors are shown in the first two plots, with their respective stability regions if they were the only environment state, and available $100\%$ of the time. The third plot shows the stability region when $\pi_1 = 0.8$ and $\pi_2 = 0.2$. This corresponds to the environment state fluctuating so that $80\%$ of the time, the service vectors from the first group are available, and $20\%$ of the time the service vectors from the second group are available to be scheduled. For any $\rho$ in the region $\mathcal{R}$ above, there is a convex combination of service modes within the resource sets which would apply a total service rate to each queue which is at least the arrival rate to that queue. For $\rho$ outside $\mathcal{R}$ there is no such combination. Service modes $S^2$ and $S^4$ are strictly dominated by a convex combination of other service vectors and therefore do not contribute to the stability region (and in fact need not be utilized to maintain stability). Service vectors with negative components such as $S^1$ and $S^5$ above may contribute to the stability region without being inside the stability region itself. The stability region for the combination of environments can be seen to be the weighted sum of the two original stability regions, with care taken to the impact of extreme points with negative components.

and proceed by contradiction. If (3.3) does not hold, then from (2.5) we must have $\lim_{t\to\infty} \frac{\int_0^t S(z)dz}{t} = \lim_{t\to\infty} \frac{\int_0^t A(z)dz}{t} + \lim_{t\to\infty} \frac{1}{t} X(0) = \rho$. But then we have

$$\begin{aligned}\rho &= \lim_{t\to\infty} \frac{\int_0^t S(z)dz}{t} \\ &= \lim_{t\to\infty} \frac{\int_0^t \sum_{e\in\mathcal{E}} \sum_{S\in\mathcal{S}^e} \mathbf{I}_{e(z)=e, S(z)=S} S dz}{t} \\ &= \sum_{e\in\mathcal{E}} \pi^e \sum_{S\in\mathcal{S}^e} \hat{\phi}^e_S S\end{aligned}$$

where $\hat{\phi} = \lim_{t\to\infty} \frac{\int_0^t \mathbf{I}_{S(z)=S} S dz}{t}$ satisfies $\hat{\phi}^e_S \geq 0$ and $\sum_{S\in\mathcal{S}^e} \hat{\phi}^e_S = 1$, which satisfies (3.2). We then easily get (arguing by contradiction) that $\limsup_{t\to\infty} X^q(t)/t > 0$ for at least one queue $q \in \mathcal{Q}$. ■

# 4 Cone Schedules and their Geometry

We focus in this paper on schedules that are workload-aware and resource-aware but not rate-aware; that is, the system's operator can observe and respond to both the environment state $e(t)$ and the workload state $X(t)$, but has no knowledge of the long-term load vector $\rho$ and state probabilities $\pi^e$.

In particular, we examine a family of resource allocation policies that are called Cone Schedules and are parameterized by a fixed matrix $\mathbf{B}$. These schedules select the service vector $\hat{S} \in \mathcal{S}^{e(t)}$ that has the maximal projection on $\mathbf{B}X(t)$, when the workload state is $X(t)$ and the environment state is $e(t) \in \mathcal{E}$. Specifically:

**Definition 4.1 (Cone Schedules)** Given a fixed $Q \times Q$ real matrix $\mathbf{B}$, a *cone schedule* is one that, when the environment is in state $e \in \mathcal{E}$ and the the workload is $X \in \mathbb{R}^Q_{0+}$, it selects a service vector $\hat{S}^e(X)$ in the set

$$\hat{\mathcal{S}}^e(X) = \arg\max_{S\in\mathcal{S}^e} \langle S, \mathbf{B}X\rangle = \{\hat{S} \in \mathcal{S}^e : \left\langle \hat{S}, \mathbf{B}X \right\rangle = \max_{S\in\mathcal{S}^e} \langle S, \mathbf{B}X\rangle\} \tag{4.1}$$

which satisfies $S_q = 0$ whenever $S_q = 0$. We show that such a vector must be contained in $\hat{\mathcal{S}}^e$ by proposition 4.1 below. The set $\hat{\mathcal{S}}^e(X) \subseteq \mathcal{S}^e$ is nonempty, but may contain several service vectors in $\mathcal{S}^e$, in which case one is arbitrarily chosen by the cone schedule. Note that

$$\left\langle \hat{S}^e(X), \mathbf{B}X \right\rangle = \max_{S\in\mathcal{S}^e} \langle S, \mathbf{B}X\rangle\,, \tag{4.2}$$

so the chosen $\hat{S}^e(X)$ is one of maximal projection on $\mathbf{B}X$ amongst those in $\mathcal{S}^e$. Therefore, the service vector $\hat{S}(t)$ chosen by the cone schedule at time $t \geq 0$ is

$$\hat{S}(t) \in \hat{\mathcal{S}}^{e(t)}(X(t)) = \arg\max_{S\in\mathcal{S}^{e(t)}} \langle S, \mathbf{B}X(t)\rangle\,,$$

based on the observed current workload $X(t)$ and environment state $e(t)$.

Notice that the maximization $\langle S, \mathbf{B}X\rangle = \sum_q S_q(\mathbf{B}X)_q$ ensures that cone schedules follow some important intuition for a scheduling rule. We see that $(\mathbf{B}X)_q$ is increasing in $X_q$ and decreasing in $X_p$ for $p \neq q$. This will increase whenever $X_q$ comes to dominate other queues. By maximizing this sum, the cone schedules all prefer large positive service rates $S_q$ whenever $(\mathbf{B}X)_q$ is large and positive. Thus the schedules will prefer remove the most workload from the longer queues, and restrict the cross-traffic added to those longer queues. The relationship to performance and load balancing is discussed in section 6.

**Proposition 4.1 (Matrices B with Negative or Zero Off-Diagonal Elements)** If the cone schedule matrix $\mathbf{B} = \{B_{ij}, i, j \in \mathcal{Q}\}$ has *negative or zero off-diagonal elements* ($\mathbf{B}_{ij} \leq 0, i \neq j$) and the service vector sets $\mathcal{S}^e$ are *complete* for each environments state $e \in \mathcal{E}$, then there must exist some $\hat{S}^e(X) \in \hat{\mathcal{S}}^e(X)$ for which we have

$$X_q = 0 \implies \hat{S}^e_q(X) \leq 0$$

for each $q \in \mathcal{Q}$. Thus, for such $\mathbf{B}$ matrices, the corresponding cone schedules can always select service vectors that provide *no positive service rate to an empty queue*.

**Proof:** Given a workload vector $X$ such that $X_q = 0$ for some (empty) queue $q \in \mathcal{Q}$, let us examine the inner product maximized by the cone schedule (4.1) in selecting $\hat{S}^e(X) \in \hat{\mathcal{S}}^e(X)$, that is,

$$\langle S, \mathbf{B}X\rangle = \sum_{i\in\mathcal{Q}}\sum_{j\in\mathcal{Q}} S_i B_{ij} X_j = \sum_{i\in\mathcal{Q}}\{S_i B_{ii} X_i + S_i(\sum_{j\in\mathcal{Q}-\{i\}} B_{ij}X_j)\}. \tag{4.3}$$

with $S \in \mathcal{S}^e$. Consider the term corresponding to the empty queue $q$ in the above sum, that is,

$$S_q B_{qq} X_q + S_q(\sum_{j\in\mathcal{Q}-\{q\}} B_{qj}X_j). \tag{4.4}$$

Since $X_q = 0$, the first term above is automatically zero, irrespectively of $S_q$ and $B_{qq}$. However, since $B_{qj} \leq 0$ and $X_j \geq 0$ for each $j \in \mathcal{Q} - \{q\}$, we see that

$$\sum_{j\in\mathcal{Q}-\{q\}} B_{qj}X_j \leq 0. \tag{4.5}$$

Arguing by contradiction, assume that $(S_1, S_2, ..., S_{q-1}, S_q > 0, S_{q+1}, ..., S_Q) \in \mathcal{S}^e$ maximizes (4.3) with $S_q > 0$. But because $\mathcal{S}^e$ is assumed to be *complete* (2.4), the vector $(S_1, S_2, ..., S_{q-1}, S_q = 0, S_{q+1}, ..., S_Q)$ also belongs to $\mathcal{S}^e$ and has $S_q = 0$, hence, leads to an equal or greater value of (4.3) because of (4.4) and (4.5). This establishes a contradiction and implies that the set of service vectors $S$ that maximize (4.3) must always include one where $S_q \leq 0$ (provide no positive service rate) for each empty queue $q \in \mathcal{Q}$ (that is, with workload $X_q = 0$).

■

To justify the term 'cone' schedule consider the following perspective. Define first the set of workloads $X$ for which the cone schedule would choose the service vector $S$ when the environment is in state $e$, that is:

$$\mathcal{C}^e_S = \left\{X \in \mathbb{R}^Q_{0+} : \langle S, \mathbf{B}X\rangle = \max_{S'\in\mathcal{S}^e} \langle S', \mathbf{B}X\rangle\right\}$$

for $S \in \mathcal{S}^e, e \in \mathcal{E}$. This is simply the set of workloads $X$ that have maximum projection on $S \in \mathcal{S}^e$ amongst all other sets in $\mathcal{S}^e$. Note that $\mathcal{C}^e_S$ is a *geometric cone* because $\langle S, \mathbf{B}X\rangle \geq \langle S', \mathbf{B}X\rangle$ implies that $\langle S, \mathbf{B}\alpha X\rangle \geq \langle S', \mathbf{B}\alpha X\rangle$ for any positive scalar $\alpha \in \mathbb{R}_+$ and $S, S' \in \mathcal{S}^e$. Thus, if $X$ belongs to $\mathcal{C}^e_S$ then any up/down-scaling $\alpha X$ also belongs to it.

For each environment state $e \in \mathcal{E}$, the cones $\mathcal{C}^e_S, S \in \mathcal{S}^e$ form a *partition* of the workload space, that is,

$$\bigcup_{S \in \mathcal{S}^e} \mathcal{C}^e_S = \mathbb{R}^Q_{0+}.$$

In general, some cones may actually be degenerate (like those corresponding to service vectors in $\mathcal{S}^e$ that are fully dominated component-wise by others in $\mathcal{S}^e$) and several cones may share common boundaries. Observe that the cone schedule can now be geometrically defined as follows:

| When the environment state is $e$ and the workload $X \in \mathcal{C}^e_S \implies$ choose $\hat{S}^e(X) = S \in \mathcal{S}^e$. |
|---|

The cone structure of the sets $\mathcal{C}^e_S$ motivates the name *cone schedules*.

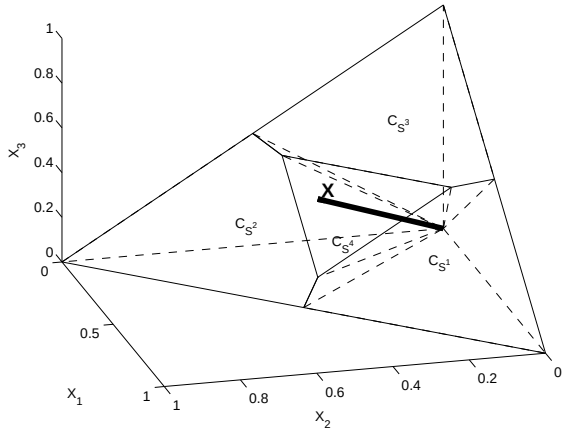



Figure 2: The cone schedules assign a service vector from $\mathcal{S}^e$ by identifying the location of $X$ with respect to the cones formed by $\mathcal{C}^e_S$. This figure shows the cone structure for a system with $Q = 3$ queues and 4 service vectors for this particular environment. When $X$ is in cone $\mathcal{C}^e_S$, then service vector $S$ corresponding to that cone is used. The vector $X$ will fluctuate within $\Re^3$, switching between service vectors when the arrivals and departures cause $X(t)$ to cross a cone boundary, or when the environment state changes. The cone boundaries are influenced by the environment state and the matrix $\mathbf{B}$.

When the environment is in state $e \in \mathcal{E}$ and the workload $X$ is in the interior of the non-degenerate cone $\mathcal{C}^e_S$, then the only service vector that can be used by the cone schedule is $S \in \mathcal{S}^e$. However, if $X$ is on the boundary of several adjacent cones (for example, $X \in \mathcal{C}^e_{S^1} \bigcap \mathcal{C}^e_{S^2} \bigcap \mathcal{C}^e_{S^3}$), then any of the service vectors corresponding to these cones can be used ($S^1$, or $S^2$, or $S^3$). Therefore, given a workload vector $X$, we want to define the cone it belongs to, which consequently specifies what service vector the cone schedule ought to use. We proceed in this direction below.

To take another perspective, recall that when the environment is in state $e \in \mathcal{E}$ and the workload is

$X$, then the cone schedule chooses a service vector $\hat{\mathcal{S}}^e(X)$ in the set

$$\hat{\mathcal{S}}^e(X) = \left\{ \hat{S} \in \mathcal{S}^e : \left\langle \hat{S}, \mathbf{B}X \right\rangle = \max_{S \in \mathcal{S}^e} \langle S, \mathbf{B}X \rangle \right\} \subseteq \mathcal{S}^e;$$

any vector is arbitrarily chosen, if there are more than one vector in the set $\hat{\mathcal{S}}^e(X)$. When $X$ is in the interior of the (non-degenerate) cone $\mathcal{C}^e_S$, then $\hat{\mathcal{S}}^e(X) = \{S\}$ is a singleton and $\hat{S}^e(X) = S$. This follows since the interior of a cone denotes all workload vectors $X$ for which the inner product $\langle S, \mathbf{B}X \rangle$ is *uniquely* maximized by $S$.

To cover the general case of $X$ being on a cone boundary (perhaps, a common boundary of several cones), we define the 'surrounding' cone of the workload vector $X$ as

$$\mathcal{C}^e(X) = \bigcup_{S \in \hat{\mathcal{S}}^e(X)} \mathcal{C}^e_S$$

For example, if $X$ is on the boundary of $\mathcal{C}^e_{S^1}$ and $\mathcal{C}^e_{S^2}$ only, then $\mathcal{C}^e(X) = \mathcal{C}^e_{S^1} \bigcup \mathcal{C}^e_{S^2}$. Note that the above definitions lead to the following equivalence

$$\mathcal{C}^e(X) \subseteq C^e(Y) \Leftrightarrow \hat{\mathcal{S}}^e(X) \subseteq \hat{\mathcal{S}}^e(Y),$$

as well as

$$\mathcal{C}^e(X) \subseteq \mathcal{C}^e(Y) \Rightarrow X \in \mathcal{C}^e(Y)$$

for any two workload vectors $X, Y \in \mathbb{R}^Q_{0+}$ and environment state $e \in \mathcal{E}$. This is illustrated in Fig. 3.

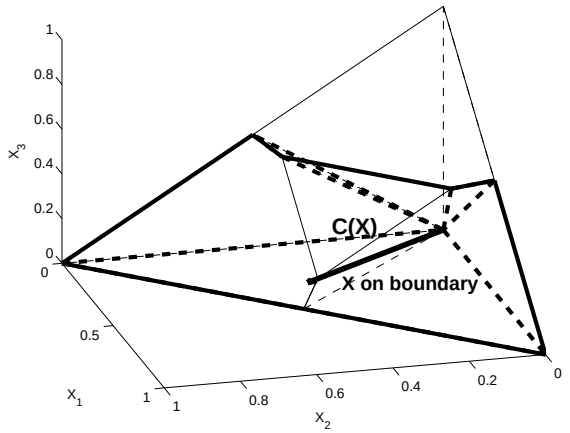



Figure 3: For workload vectors $X$ which lie precisely on the boundary of two or more cones, the cone $\mathcal{C}^e(X)$ is the union of all of the cones in $\mathcal{C}^e$ which include $X$. In contrast to Fig. 2, where $X$ was interior to a single cone, the above illustration shows $X$ at the boundary of 3 of the $\mathcal{C}_S$ cones. In this case $\mathcal{C}(X) = \mathcal{C}_{S^1} \cup \mathcal{C}_{S^2} \cup \mathcal{C}_{S^4}$ includes all the elements of the three different cones. This definition is important in the proof because it captures the workload vectors which share an optimal service vector with $X$.

Note that if $Y \in \mathcal{C}^e(X)$ then there must exist a service vector $\hat{S} \in \mathcal{S}^e$ for which both $\langle S, \mathbf{B}X \rangle$ is maximized at $\hat{S}$ and $\langle S, \mathbf{B}Y \rangle$ is maximized at $\hat{S}$, and if $Y \notin \mathcal{C}^e(X)$ then no such vector can exist.

We observe that $X$ cannot be on an interior boundary of $\mathcal{C}^e(X)$ (the only boundary it could be on is where the cone meets an axis because of the non-negativity constraint). If $X$ were on an interior boundary

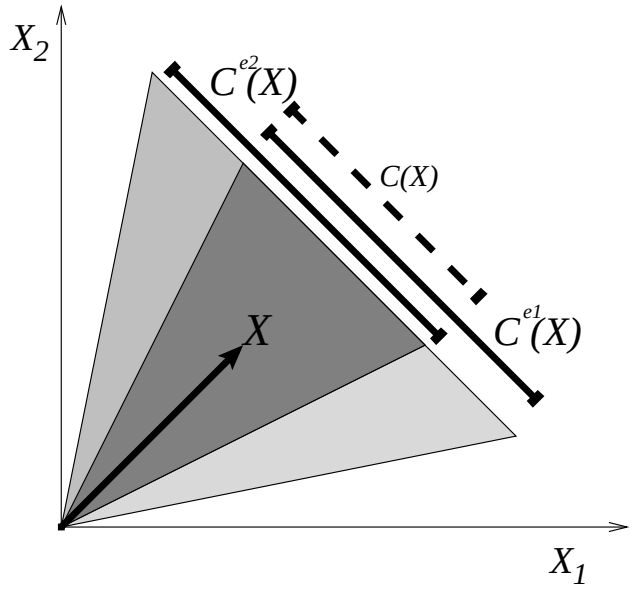



Figure 4: The cone $\mathcal{C}(X)$ over environments $\mathcal{E}$ is illustrated. Here, $X$ is in the cones $\mathcal{C}^{e_1}(X)$ and $\mathcal{C}^{e_2}(X)$ for the two environments $e_1$ and $e_2$. The cone $\mathcal{C}(X)$ is the intersection of both of those cones. Since $X$ is known to be on the interior of each cone, $X$ is also on the interior of $\mathcal{C}(X)$.

then there must exist a direction vector $\delta \neq 0$ for which $(X + \lambda\delta) \geq 0$ and $(X + \lambda\delta) \notin \mathcal{C}^e(X)$ for an arbitrarily small positive scalar $\lambda$. This means that there exists some service vector $S^\delta \in \mathcal{S}^e$ for which $\left\langle S^\delta, \mathbf{B}(X + \lambda\delta)\right\rangle > \langle S, \mathbf{B}(X + \lambda\delta)\rangle$ for all $S \in \hat{\mathcal{S}}^e$. But since $S^\delta \notin \hat{\mathcal{S}}^e(X)$ we also have $\left\langle S^\delta, \mathbf{B}X\right\rangle < \langle S, \mathbf{B}X\rangle$ for all $S \in \hat{\mathcal{S}}^e$. This leads to the inequality

$$\lambda(\left\langle S^\delta, \mathbf{B}\delta\right\rangle - \langle S, \mathbf{B}\delta\rangle) > \langle S, \mathbf{B}X\rangle - \left\langle S^\delta \mathbf{B}X\right\rangle > 0.$$

Since the left hand side can be made arbitrarily small this leads directly to a contradiction and we conclude that $X$ is indeed on the strict interior of $\mathcal{C}^e(X)$. This observation becomes critical in the proof of stability.

Finally, we define the cone around $X$ with respect to *all* environment states $e \in \mathcal{E}$ as

$$\mathcal{C}(X) = \bigcap_{e \in \mathcal{E}} \mathcal{C}^e(X).$$

The cone $\mathcal{C}(X)$ is illustrated in Fig 4 is of course non-empty because $X$ belongs to each cones $\mathcal{C}^e(X), e \in \mathcal{E}$. This is the cone of workloads $Y$ for which, at each environment state $e \in \mathcal{E}$, the cone schedule could have selected for $Y$ the same service vector as for $X$ (fixed), that is,

$$\mathcal{C}(X) = \{Y \in \mathbb{R}^Q_{0+} : \hat{\mathcal{S}}^e(Y) \bigcap \hat{\mathcal{S}}^e(X) \neq \emptyset, \text{ for each } e \in \mathcal{E}\};$$

hence, when $Y \in \mathcal{C}(X)$, then for each $e \in \mathcal{E}$ we have $\hat{S}^e(Y) \in \hat{\mathcal{S}}^e(X)$, besides $\hat{S}^e(X) \in \hat{\mathcal{S}}^e(X)$ of course. Viewed another way,

$$\mathcal{C}(X) = \{Y \in \mathbb{R}^Q_{0+} : \left\langle \hat{S}^e(Y), \mathbf{B}X \right\rangle = \max_{S \in \mathcal{S}^e} \langle S, \mathbf{B}X \rangle \text{ for each } e \in \mathcal{E}\},$$

that is, when $Y \in \mathcal{C}(X)$, then for each $e \in \mathcal{E}$ we have that $\hat{S}^e(Y)$ has maximal projection on $\mathbf{B}X$, besides also having maximal projection on $\mathbf{B}Y$ (by definition).

We note that since $X$ is strictly on the interior of each cone $\mathcal{C}^e(X)$ and there are finitely many environment states in $\mathbf{E}$ then $X$ is strictly on the interior of $\mathcal{C}(X)$.

The cone $\mathcal{C}(X)$ turns out to be of key importance in the stability proof below. This completes the geometric picture of cone schedules.

# 5 Universal Stability of Cone Schedules

We are primarily interested in the throughput maximizing properties of cone schedules for various families of matrices $\mathbf{B}$, given the traffic load $\rho$. The following theorem establishes that stability can be maintained for any $\rho \in \mathcal{R}$ by rich families of matrices $\mathbf{B}$.

Consider a cone schedule generated by the matrix $\mathbf{B}$ and operating on any arbitrarily fixed system $\Sigma$ chosen from the class $\mathfrak{S}$ of processing systems defined by:

1. some set of queues $\mathcal{Q}$ and some set of environment states $\mathcal{E}$,
2. some environment trace $\mathbf{E} \in \mathfrak{E}(\pi^e, e \in \mathcal{E})$, as per (2.3),
3. some (non-empty) service vector sets $\mathcal{S}^e, e \in \mathcal{E}$ that are *complete*, as per (2.4),
4. some traffic trace $\mathbf{A} = \{A(t), t \geq 0\} \in \mathfrak{A}(\rho)$ with load $\rho(\mathbf{A}) = \rho$, as per (2.2).

**Theorem 5.1 (Universal Stability of Cone Schedules)** Given the above assumptions if $\mathbf{B}$ is positive-definite, symmetric and has negative or zero off-diagonal elements ($B_{qp} \leq 0, p \neq q \in \mathcal{Q}$), then

$$\rho(\mathbf{A}) \in \mathcal{R}(\mathcal{S}^e, \pi^e, e \in \mathcal{E}) \implies \lim_{t \to \infty} \frac{X(t)}{t} = 0 \tag{5.1}$$

*universally* on $\mathfrak{S}$ . That is, each system in $\mathfrak{S}$ is (rate) stable under such a cone schedule, when $\rho(\mathbf{A}) \in \mathcal{R}(\mathcal{S}^e, \pi^e, e \in \mathcal{E})$.

It turns out that $\mathbf{B}$ being positive definite and having nonpositive off-diagonal elements are both necessary for universal stability, which was shown in [Ross and Bambos, 2009].

To see why nonpositive off diagonal elements are required, consider a simple network with $Q = 2$ queues and $E = 1$ environment state, where $\mathbf{B} = [2, 1; 1, 2]$ is used. If $S^1 = (1, 0)$ and $S^2 = (0, 3)$ are the two available service vectors then $\left\langle S^1, \mathbf{B}X \right\rangle = 2X_1 + X_2$, and $\left\langle S^2, \mathbf{B}X \right\rangle = 3X_1 + 6X_2$. Since $\left\langle S^2, \mathbf{B}X \right\rangle$ strictly dominates $\left\langle S^2, \mathbf{B}X \right\rangle$ for any nonzero workload, $S^1$ would never be selected and any arrival process with $\rho_1 > 0$ will be unstable.

To see why positive definiteness is required, consider a simple network with $Q = 2$ queues and $E = 1$ environment state, where $\mathbf{B} = [1, -2; -2, 1]$ is used. Let $S^1 = (1, 1), S^2 = (0, 0), S^3 = (1, 0)$ and $S^4 = (0, 1)$ be the available service vectors. Then we have $\left\langle S^1, \mathbf{B}X \right\rangle = -X_1 - X_2 < 0 = \left\langle S^2, \mathbf{B}X \right\rangle$, and $S^1$ will never be selected. The effective service rates applied to the queues $\hat{S}_q = \lim_{t\to\infty} \frac{1}{t} \int_o^t S_q(z)$ must then satisfy $\hat{S}_1 + \hat{S}_2 \leq 1$. Now $\rho = (\rho_1, \rho_2)$ with $0.5 < \rho_q \leq 1$ is contained within $\mathcal{R}$ by (3.2), but rate stability cannot possibly be achieved in (2.5). The parameters of the non-positive-definite matrix $\mathbf{B}$ cause the cone schedule to avoid utilizing $S^1$, which is critical for rate stability because it lies on the convex hull of $\mathcal{R}$.

### 5.1 Proof of Theorem 5.1

We prove rate stability via a sequence of intermediate steps.

Consider any arbitrarily fixed environment trace $\mathbf{E} = \{e(t), t \geq 0\}$, such that $\mathcal{S}^e$ is *complete* and

$$\lim_{t\to\infty} \frac{\int_0^t \mathbf{1}_{\{e(z)=e\}} dz}{t} = \pi^e$$

for each $e \in \mathcal{E}$. Consider also any arbitrarily fixed traffic trace $\mathbf{A} = \{A(t), t \geq 0\}$ satisfying

$$\lim_{t\to\infty} \frac{\int_0^t A(z)dz}{t} = \rho(\mathbf{A}) \in \mathcal{R}(\mathcal{S}^e, \pi^e, e \in \mathcal{E}).$$

We note that while $\mathbf{A}$ and $\mathbf{E}$ are fixed, they can be generated arbitrarily, including by an underlying stochastic process or an adversary. Recall that by Proposition (4.1) when $\mathbf{B}$ has negative or zero off-diagonal elements the generated cone schedule applies no positive rate to empty queues. Therefore,

$$X(t) = X(0) + \int_0^t A(z)dz - \int_0^t S(z)dz \tag{5.2}$$

for the workload $X(t)$ at time $t$ – as in (2.5) – without having to compensate for any idle time.

**Proposition 5.1** Under the conditions conditions of Theorem 5.1, the service vectors $\hat{S}^e(X) \in \hat{\mathcal{S}}^e(X)$ selected by the cone schedule under various environment states $e \in \mathcal{E}$ satisfy

$$\langle \rho, \mathbf{B}X \rangle \leq \sum_{e\in\mathcal{E}} \pi^e \left\langle \hat{S}^e(X), \mathbf{B}X \right\rangle = \sum_{e\in\mathcal{E}} \pi^e \max_{S\in\mathcal{S}^e} \langle S, \mathbf{B}X \rangle . \tag{5.3}$$

for each workload $X \in \mathbb{R}^Q_{0+}$.

**Proof:** First, choose any workload $X$ and fix it. Since $\rho \in \mathcal{R}$, we have $\rho \leq \sum_{e\in\mathcal{E}} \pi^e \sum_{S\in\mathcal{S}^e} \phi^e_S S$ according to (3.2), or

$$0 \leq \rho_q \leq \sum_{e\in\mathcal{E}} \pi^e \sum_{S\in\mathcal{S}^e} \phi^e_S S_q, \text{ for each } q \in \mathcal{Q}, \tag{5.4}$$

for some positive weights $\phi^e_S \geq 0$ such that $\sum_{S\in\mathcal{S}^e} \phi^e_S \leq 1$.

We denote $v_q = (\mathbf{B}X)_q$ and note that this may be negative for some $q \in \mathcal{Q}$. We examine, the following two cases:

1. If $v_q = (\mathbf{B}X)_q \geq 0$, we get from (5.4) that

$$\rho_q v_q \leq \sum_{e\in\mathcal{E}} \pi^e \sum_{S\in\mathcal{S}^e} \phi_S^e S_q v_q. \tag{5.5}$$

2. If $v_q = (\mathbf{B}X)_q < 0$, we have

$$\rho_q v_q \leq 0,$$

since $\rho_q \geq 0$.

Combining the two cases, we get

$$\rho_q v_q \leq \sum_{e\in\mathcal{E}} \pi^e \sum_{S\in\mathcal{S}^e} \phi_S^e S_q \mathbf{1}_{\{v_q S_q \geq 0\}} v_q \leq \sum_{e\in\mathcal{E}} \pi^e \sum_{S\in\mathcal{S}^e} \phi_S^e S_q \mathbf{1}_{\{v_q \geq 0\}} \mathbf{1}_{\{s_q > 0\}} v_q$$

for $q \in \mathcal{Q}$. Adding the terms up over $q \in \mathcal{Q}$, we get

$$\langle \rho, v\rangle \leq \sum_{e\in\mathcal{E}} \pi^e \sum_{S\in\mathcal{S}^e} \phi_S^e \langle V(S), v\rangle \tag{5.6}$$

where $V(S) = (S_q \mathbf{1}_{\{v_q\geq 0, S_q>0\}}), q \in \mathcal{Q})$ is the vector generated by the service vector $S \in \mathcal{S}^e$ by setting 0 the components $S_q$ for which $v_q < 0$ and $S_q > 0$.

Now recall that for each $e \in \mathcal{E}$ and $S \in \mathcal{S}^e$, $V(S)$ is a *sub-vector* of $S$ (dropping some positive components to 0) and is also in $\mathcal{S}^e$ because the latter set is *complete*. But the service vector $\hat{S}^e(X)$ selected by the cone schedule (4.2) has the maximal projection on $v = \mathbf{B}X$ amongst all those in $\mathcal{S}^e$, so $\langle V(S), v\rangle \leq \left\langle \hat{S}^e(X), v \right\rangle$ for every $S \in \mathcal{S}^e$. Therefore, (5.6) becomes

$$\langle \rho, v\rangle \leq \sum_{e\in\mathcal{E}} \pi^e \sum_{S\in\mathcal{S}^e} \phi_S^e \langle V(S), v\rangle \leq \sum_{e\in\mathcal{E}} \pi^e \left[\sum_{S\in\mathcal{S}^e} \phi_S^e\right] \left\langle \hat{S}^e(X), v \right\rangle \leq \sum_{e\in\mathcal{E}} \pi^e \left\langle \hat{S}^e(X), v \right\rangle,$$

where the last inequality holds because $\sum_{S\in\mathcal{S}^e} \phi_S^e \leq 1$ for each $e \in \mathcal{E}$. Putting back $v = \mathbf{B}X$, we get

$$\langle \rho, \mathbf{B}X\rangle \leq \sum_{e\in\mathcal{E}} \pi^e \left\langle \hat{S}^e(X), \mathbf{B}X \right\rangle,$$

which completes the proof.

■

**Lemma 5.1** We have that $\lim_{t\to\infty} \frac{X(t)}{t} = 0$ implies $\lim_{t\to\infty} \frac{\int_0^t \hat{S}(z)dz}{t} = \rho$. That is, the long-term applied service rate is equal to the long-term traffic load, when the system is (rate) stable.

**Proof:** This is immediately obtained by dividing (5.2) by $t$ and letting $t\to\infty$. ■

**Lemma 5.2** Consider two arbitrarily fixed, increasing, unbounded time sequences $\{t_n\}_{n=1}^\infty$ and $\{s_n\}_{n=1}^\infty$ with $s_n \leq t_n$ for each $n \geq 1$. If $\lim_{n\to\infty} \frac{t_n - s_n}{t_n} = 0$ (or equivalently $\lim_{n\to\infty} \frac{s_n}{t_n} = 1$), then

$$\lim_{n\to\infty} \frac{X(t_n) - X(s_n)}{t_n} = \lim_{n\to\infty} \frac{X(t_n) - X(s_n)}{s_n} = 0.$$

**Proof:** Note that $0 \leq \int_{s_n}^{t_n} \mathbf{1}_{\{\hat{S}(z)=S\}} dz \leq t_n - s_n$ for each $S \in \mathcal{S} = \cup_{e \in \mathcal{E}} \mathcal{S}^e$. Dividing by $t_n$ and taking the limit as $n \to \infty$, we get $\lim_{n\to\infty} \frac{\int_{s_n}^{t_n} \mathbf{1}_{\{\hat{S}(z)=S\}} dz}{t_n} = 0$. Recalling that

$$X(t_n) - X(s_n) = \int_{s_n}^{t_n} A(z)dz - \sum_{S \in \mathcal{S}} S \int_{s_n}^{t_n} \mathbf{1}_{\{\hat{S}(z)=S\}} dz,$$

dividing by $t_n$ and letting $n \to \infty$, we get

$$\begin{aligned}
\lim_{n\to\infty} \frac{X(t_n) - X(s_n)}{t_n} &= \lim_{n\to\infty} \frac{\int_{s_n}^{t_n} A(z)dz}{t_n} - \lim_{n\to\infty} \frac{\sum_{S\in\mathcal{S}} S \int_{s_n}^{t_n} \mathbf{1}_{\{\hat{S}(z)=S\}} dz}{t_n} \\
&= \lim_{n\to\infty} \frac{\int_0^{t_n} A(z)dz}{t_n} - \lim_{n\to\infty} \frac{\int_0^{s_n} A(z)dz}{s_n} \frac{s_n}{t_n} - 0 \\
&= \rho_q - \rho_q.1 \\
&= 0
\end{aligned} \tag{5.7}$$

Moreover, $\lim_{n\to\infty} \frac{X(t_n)-X(s_n)}{s_n} = \lim_{n\to\infty} \frac{X(t_n)-X(s_n)}{t_n} \frac{t_n}{s_n} = 0$. This completes the proof. ■

**Lemma 5.3** Consider an arbitrarily fixed, increasing, unbounded time sequence $\{t_n\}_{n=1}^{\infty}$. The following result then holds:

$$\lim_{n\to\infty} \frac{X(t_n) - X(t_n^-)}{t_n} = 0.$$

**Proof:** Clearly the result holds at times $t$ when $A(t)$ is finite. The issue arises at times $t_n$ when $A(t_n)$ has a $\delta$-jump and the workload suddenly shifts by a finite amount, which may actually be increasing in consecutive jumps. Let $t_n$ be the time of a job arrival to queue $q \in \mathcal{Q}$, where $j_n$ the index of that job and $\sigma_q^{j_n}$ the workload added by the job. It is then sufficient to show that $\lim_{n\to\infty} \frac{\sigma_q^{j_n}}{t_n} = 0$. Indeed, note that

$$\sigma_q^{j_n} = \int_0^{t_n} A_q(t)dt - \int_0^{t_n^-} A_q(t)dt \tag{5.8}$$

Dividing by $t_n$ and letting $n \to \infty$, we have $\lim_{n\to\infty} \frac{\sigma_q^{j_n}}{t_n} = \rho_q - \rho_q = 0$, which proves the lemma. ■

### 5.1.1 Building a Contradiction.

The objective of the proof is to show that $\lim_{t\to\infty} \frac{X(t)}{t} = 0$, when $\rho \in \mathcal{R}$. Since $\mathbf{B}$ is a *positive-definite* matrix, it is sufficient to show that $\lim_{t\to\infty} \left\langle \frac{X(t)}{t}, \mathbf{B}\frac{X(t)}{t} \right\rangle = 0$.

The proof proceeds by contradiction. Assume that $\limsup_{t\to\infty} \left\langle \frac{X(t)}{t}, \mathbf{B}\frac{X(t)}{t} \right\rangle > 0$, and let $\{t_a\}_{a=1}^{\infty}$ be an increasing unbounded time sequence on which the supremum limit is obtained; let

$$\lim_{a\to\infty} \frac{X(t_a)}{t_a} = \eta \neq 0 \tag{5.9}$$

be the corresponding limit. Such a convergent subsequence must exist by the compactness (since bounded) of the set of possible values[7] for $\frac{X(t)}{t}$ at large times. We will construct a related unbounded time sequence

[7]For any arrival trace, we have $X(t) \leq \int_0^t A(s)ds$, which implies that $\frac{X(t)}{t} \leq \frac{\int_0^t A(s)ds}{t} \to \rho$

$\{s_d\}_{d=1}^{\infty}$ and show that it has the property $\lim_{d\to\infty}\left\langle \frac{X(s_d)}{s_d}, \mathbf{B}\frac{X(s_d)}{s_d}\right\rangle > \lim_{a\to\infty}\left\langle \frac{X(t_a)}{t_a}, \mathbf{B}\frac{X(t_a)}{t_a}\right\rangle > 0$. The existence of such a sequence will *contradict* that the supremum limit is attained on $\{t_a\}_{a=1}^{\infty}$.

We establish the required contradiction by finding an increasing unbounded subsequence $\{t_c\}_{c=1}^{\infty}$ of $\{t_a\}_{a=1}^{\infty}$, and a related sequence $\{s_c\}_{c=1}^{\infty}$, which satisfy the following two *Key Properties*:

**I.** $\lim_{c\to\infty}\frac{t_c-s_c}{t_c}=\epsilon\in(0,1)$ and $s_c<t_c$ for each $c$. This implies that $\lim_{c\to\infty}\frac{s_c}{t_c}=1-\epsilon$.

**II.** $\mathcal{C}(X(t))\subset\mathcal{C}(\eta)$ for all $t\in(s_c,t_c]$ and each $c$. This implies that the workload $X(t)$ drifts within the cone $\mathcal{C}(\eta)$ surrounding $\eta=\lim_{c\to\infty}\frac{X(t_c)}{t_c}$ throughout the time interval $(s_c,t_c]$.

The associated *intuition* is that $s_c$ marks the last time before $t_c$ that the workload vector $X(s_c)$ (re)enters the cone $\mathcal{C}(\eta)$ and reaches $X(t_c)\approx\eta t_c$ at time $t_c$, drifting in $\mathcal{C}(\eta)$ throughout the time interval $(s_c,t_c]$.

Before constructing the above sequences with properties I and II we show their implications for establishing the required contradiction.

**Lemma 5.4** If the sequences $\{t_c\}_{c=1}^{\infty}$ and $\{s_c\}_{c=1}^{\infty}$ satisfy the *Properties I and II* above, then the supremum limit is not attained - as initially assumed - on the sequence $\{t_c\}_{c=1}^{\infty}$ (which is a subsequence of $\{t_a\}_{a=1}^{\infty}$). This establishes the targeted contradiction.

**Proof:** Since $\mathbf{B}$ matrix has negative or zero off diagonal elements, the cone schedule does not apply any positive service rate to any empty queue (4.1). Therefore, by (5.2) we have

$$X(t_c)-X(s_c)=\int_{s_c}^{t_c}A(z)dz-\int_{s_c}^{t_c}\hat{S}(z)dz \tag{5.10}$$

and projecting on $\mathbf{B}\eta$ (5.9) we get

$$\begin{aligned}\langle X(t_c)-X(s_c),\mathbf{B}\eta\rangle &= \left\langle\int_{s_c}^{t_c}A(z)dz,\mathbf{B}\eta\right\rangle-\left\langle\int_{s_c}^{t_c}\hat{S}(z)dz,\mathbf{B}\eta\right\rangle\\ &= \left\langle\int_{s_c}^{t_c}A(z)dz,\mathbf{B}\eta\right\rangle-\sum_{e\in\mathcal{E}}\int_{s_c}^{t_c}\left\langle\hat{S}^e(X(z)),\mathbf{B}\eta\right\rangle\mathbf{1}_{\{e(z)=e\}}dz \end{aligned}\tag{5.11}$$

where $\hat{S}^e(X(z))\in\hat{\mathcal{S}}^e(X(z))$ for $z\geq 0$. But because of Property II above, the workload $X(z)$ drifts in the cone $\mathcal{C}(\eta)$ throughout $z\in(s_c,t_c]$, which implies that

$$\left\langle\hat{S}^e(X(z)),\mathbf{B}\eta\right\rangle=\max_{S\in\mathcal{S}^e}\langle S,\mathbf{B}\eta\rangle$$

when $e(z)=e\in\mathcal{E}$. Substituting into (5.11) we get

$$\langle X(t_c)-X(s_c),\mathbf{B}\eta\rangle=\left\langle\int_{s_c}^{t_c}A(z)dz,\mathbf{B}\eta\right\rangle-\sum_{e\in\mathcal{E}}\left(\int_{s_c}^{t_c}\mathbf{1}_{\{e(z)=e\}}dz\right)\max_{S\in\mathcal{S}^e}\langle S,\mathbf{B}\eta\rangle \tag{5.12}$$

Observe now that

$$\begin{aligned}\lim_{c\to\infty}\frac{\int_{s_c}^{t_c}A(t)dt}{t_c-s_c} &= \lim_{c\to\infty}\frac{\int_0^{t_c}A(t)dt}{t_c}\lim_{c\to\infty}\frac{t_c}{t_c-s_c}-\lim_{c\to\infty}\frac{\int_0^{s_c}A(t)dt}{s_c}\lim_{c\to\infty}\frac{s_c}{t_c-s_c}\\ &= \rho\frac{1}{\epsilon}-\rho(\frac{1}{\epsilon}-1)\\ &= \rho\end{aligned}$$

because of Property I above. Dividing 5.12 by $(t_c - s_c)$ and letting $c \to \infty$, we get

$$\lim_{c\to\infty} \left\langle \frac{X(t_c) - X(s_c)}{t_c - s_c}, \mathbf{B}\eta \right\rangle = \langle \rho, \mathbf{B}\eta \rangle - \sum_{e\in\mathcal{E}} \pi^e \max_{S\in\mathcal{S}^e} \langle S, \mathbf{B}\eta \rangle = -\gamma(\eta) \leq 0 \tag{5.13}$$

for $\gamma(\eta) \geq 0$. The inequality $-\gamma(\eta) = \langle \rho, \mathbf{B}\eta \rangle - \sum_{e\in\mathcal{E}} \pi^e \max_{S\in\mathcal{S}^e} \langle S, \mathbf{B}\eta \rangle \leq 0$ is due to (5.3), since it is assumed that $\rho \in \mathcal{R}$.

Since $\{t_c\}_{c=1}^{\infty}$ is a subsequence of $\{t_a\}_{a=1}^{\infty}$ we have $\lim_{c\to\infty} \frac{X(t_c)}{t_c} = \eta$. Using Property I and (5.13) we get the following inequality

$$\begin{aligned}
\lim_{c\to\infty} \left\langle \frac{X(s_c)}{s_c}, \mathbf{B}\eta \right\rangle &= \lim_{c\to\infty} \left\{ \left\langle \frac{X(s_c) - X(t_c)}{s_c}, \mathbf{B}\eta \right\rangle + \left\langle \frac{X(t_c)}{s_c}, \mathbf{B}\eta \right\rangle \right\} \\
&= \lim_{c\to\infty} \left\{ \frac{t_c - s_c}{s_c} \left[ - \left\langle \frac{X(t_c) - X(s_c)}{t_c - s_c}, \mathbf{B}\eta \right\rangle \right] + \frac{t_c}{s_c} \left\langle \frac{X(t_c)}{t_c}, \mathbf{B}\eta \right\rangle \right\} \\
&= \frac{\epsilon}{1-\epsilon}\, \gamma(\eta) + \frac{1}{1-\epsilon}\, \langle \eta, \mathbf{B}\eta \rangle \\
&> \langle \eta, \mathbf{B}\eta \rangle
\end{aligned} \tag{5.14}$$

The last inequality is due to the facts that $\epsilon \in (0, 1)$ and $\gamma(\eta) \geq 0$.

By successive thinnings of the components of the workload vector, we can obtain an increasing unbounded subsequence $\{s_d\}_{d=1}^{\infty}$ of $\{s_c\}_{c=1}^{\infty}$ such that $\lim_{d\to\infty} \frac{X(s_d)}{s_d} = \psi$ and from (5.14)

$$\langle \psi, \mathbf{B}\eta \rangle > \langle \eta, \mathbf{B}\eta \rangle \tag{5.15}$$

Since $\mathbf{B}$ is *positive-definite* we have $\langle \psi - \eta, \mathbf{B}(\psi - \eta) \rangle \geq 0$. This implies $\langle \psi, \mathbf{B}\psi \rangle + \langle \eta, \mathbf{B}\eta \rangle \geq \langle \psi, \mathbf{B}\eta \rangle + \langle \eta, \mathbf{B}\psi \rangle$. Since $\mathbf{B}$ is *symmetric* (self-adjoint) we have $\langle \eta, \mathbf{B}\psi \rangle = \langle \psi, \mathbf{B}\eta \rangle$. Therefore,

$$\langle \psi, \mathbf{B}\psi \rangle + \langle \eta, \mathbf{B}\eta \rangle \geq 2 \langle \psi, \mathbf{B}\eta \rangle > 2 \langle \eta, \mathbf{B}\eta \rangle ,$$

using (5.15) for the last inequality. Thus, $\langle \psi, \mathbf{B}\psi \rangle > \langle \eta, \mathbf{B}\eta \rangle$ or

$$\lim_{d\to\infty} \left\langle \frac{X(s_d)}{s_d}, \mathbf{B}\frac{X(s_d)}{s_d} \right\rangle = \langle \psi, \mathbf{B}\psi \rangle > \langle \eta, \mathbf{B}\eta \rangle = \limsup_{t\to\infty} \left\langle \frac{X(t)}{t}, \mathbf{B}\frac{X(t)}{t} \right\rangle > 0,$$

giving a contradiction to the definition of $\eta$. This completes the proof of Lemma 5.4. ■

### 5.1.2 Constructing Sequences with Properties I and II

It now remains to construct sequences $\{t_c\}_{c=1}^{\infty}$ and $\{s_c\}_{c=1}^{\infty}$ satisfying properties I and II. Their construction is based on the intuition mentioned above, which is made formal in the following lemma.

**Lemma 5.5** Suppose $\lim_{k\to\infty} \frac{X(t_k)}{t_k} = \eta \neq 0$ for some increasing unbounded sequence $\{t_k\}_{k=1}^{\infty}$ and nonzero $\eta$. Let

$$s_k = \sup\{t < t_k : \mathcal{C}(X(t)) \not\subseteq \mathcal{C}(\eta)\} \tag{5.16}$$

be the last time before $t_k$ that the cone $\mathcal{C}(X(t))$ is not included in $\mathcal{C}(\eta)$. This is the last time that $X(t)$ crosses from outside $\mathcal{C}(\eta)$ to inside, hence, $X(t) \in \mathcal{C}(\eta)$ for every $t \in (s_k, t_k]$ and the workload drifts in

$\mathcal{C}(\eta)$ throughout that interval. By convention $s_k = 0$ if the workload has always been in $\mathcal{C}(\eta)$ before $t_k$. We then have

$$\liminf_{k\to\infty} \frac{t_k - s_k}{t_k} = \epsilon_1 > 0 \tag{5.17}$$

for some $\epsilon_1 \in (0, 1)$.

**Proof:** Arguing by contradiction, suppose that there exists an increasing unbounded subsequence $\{t_n\}_{n=1}^{\infty}$ of $\{t_k\}_{k=1}^{\infty}$ such that $\lim_{n\to\infty} \frac{t_n - s_n}{t_n} = 0$. From Lemma 5.2 we have that $\lim_{n\to\infty} \frac{X(t_n) - X(s_n)}{s_n} = 0$. Since $\lim_{n\to\infty} \frac{X(t_n)}{t_n} = \eta$, we then get $\lim_{n\to\infty} \frac{X(s_n)}{s_n} = \eta$. Further, to allow for the possibility of job arrival that instantaneously shifts the workload from outside $\mathcal{C}(\eta)$ to inside, we note from Lemma 5.3 that we have $\lim_{n\to\infty} \frac{X(s_n^-)}{s_n} = \eta$. But according to the definition of $s_n$ the workload $X(s_n)$ must be outside $\mathcal{C}(\eta)$, so $\lim_{n\to\infty} \frac{X(s_n^-)}{s_n}$ could not converge to $\eta$. This establishes the necessary contradiction, showing 5.17 and completing the proof of the Lemma 5.5. ■

We are now ready to construct sequence $\{s_c\}_{c=1}^{\infty}$ satisfying properties I and II. We rename the sequence defined in (5.16) to be $\{\hat{s}_c\}$ and choose $s_c = \max\{\hat{s}_c, (1-\epsilon_2)t_c\}$, for some $\epsilon_2 \in (0, 1)$. (The second term $(1-\epsilon_2)t_c$ is used to guard against the degenerate case where $\hat{s}_c$ is finite because the workload $X(t)$ is always in $\mathcal{C}(\eta)$ after some finite time.) Then we have the properties:

1. $\lim_{c\to\infty} \frac{t_c - s_c}{t_c} = \epsilon \in (0, 1)$ and $s_c < t_c$ for each (large) $c$.
2. $C(X(t)) \subset C(\eta)$ for all $t \in (s_c, t_c]$ and each (large) $c$.

This means that $\{s_c\}_{c=1}^{\infty}$ and $\{t_c\}_{c=1}^{\infty}$ satisfy both Properties I and II, and Lemma 5.4 completes the proof of rate stability in Theorem 5.1. ■

# 6 Performance Issues

Section 3 established the universal stability of cone schedules for an entire class of matrices $\mathbf{B}$. A natural question to consider is how the selection of $\mathbf{B}$ from within this class of matrices will affect other performance measures such as average workload and waiting time.

In [Stolyar, 2004] it was shown that for a similar queueing system (with one environment state and nonnegative $S_q$ vectors), the class of *MaxWeight* schedules, which are equivalent to cone schedules with a diagonal $\mathbf{B}$ matrix, will minimize the total workload in the system as well as the holding cost rate asymptotically in heavy traffic. While a formal proof of a corresponding result is beyond the scope of this paper, we conjecture that a similar result will hold for these generalized cone schedules. This can be observed by considering the limiting behavior of the cone schedules when $X$ is large.

$$\begin{aligned}\left\langle X(t^+), \mathbf{B}, X(t^+)\right\rangle = \\ \left\langle X(t), \mathbf{B}, X(t)\right\rangle + \left\langle A(t) - S(t), \mathbf{B}(A(t) - S(t))\right\rangle + 2\left\langle A(t), \mathbf{B}X(t)\right\rangle - 2\left\langle S(t), \mathbf{B}X(t)\right\rangle\end{aligned} \tag{6.1}$$

for $t^+ > t$ the workload immediately after time $t$. If $X(t)$ is large and fixed, then minimizing the expectation of the above equation is equivalent to maximizing $\langle X(t), \mathbf{B}S(t)\rangle$, because the first term is fixed

by $X(t)$ and no other terms grow with $X(t)$. This causes us to conjecture that the schedule that minimizes $\lim_{t\to\infty} \frac{1}{t} \int_0^t \langle X(z), \mathbf{B}X(z)\rangle$ will be the cone schedule with matrix $\mathbf{B}$. A full proof of this optimality would require considerably more restriction on the arrival trace than has been presented in this paper.

From a geometric point of view, the cones $\mathcal{C}_m^e$ shift (and expand or contract), as the weights assigned to particular queues are adjusted. Figure 5 illustrates a simple system where the matrix $\mathbf{B}$ transforms the cone space. The diagonal elements of $\mathbf{B}$ expand and contract the cones in the dimension of the corresponding queue. The off-diagonal elements move the boundary between adjacent cones where both cones have a nonzero service rate to the two corresponding queues.

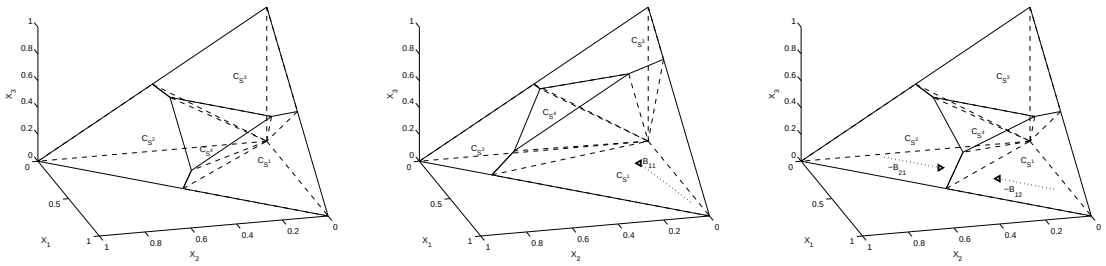


Figure 5: **The cone space with varying $\mathbf{B}$ matrices**. The plots above illustrate the impact of matrix $\mathbf{B}$ on the cone space for a system with $Q = 3$ queues and $M = 4$ service vectors. The first plot shows the cones for an identity matrix $\mathbf{B}$. The middle plot has a diagonal $\mathbf{B}$ with large positive weight on the first queue, focusing more attention on service vectors which serve that queue. The third plot shows the impact of off diagonal elements of $\mathbf{B}$. These elements affect the boundary between two queues.

As highlighted in the stability proof, as $X$ increases in magnitude in any given direction, the service vectors applied will rotate through the vectors which maximize $\langle S, \mathbf{B}X\rangle$ for each environment. While in a particular environment, $X$ will be drawn toward the boundary of its current cone. In particular, if no new arrivals were allowed and the environment stayed constant, then $X$ would follow a deterministic path to a cone boundary. Once at a boundary, the cone schedule would fluctuate around the service vectors which are optimal at that boundary and the workload would be drawn down along that boundary. The boundary planes act as attractors for the balance of queues in the system. Therefore one can view the matrix $\mathbf{B}$ as transforming the cone space in order to set the appropriate attracting boundary planes given by the cone intersections. This leads to an understanding of $\mathbf{B}$ as an important control on the relative importance of different workload dimensions.

Cone schedules perform constrained *dynamic load balancing* of the queue workloads (weighted by the elements of $\mathbf{B}$), observing the service constraints encoded in the service vectors $\mathcal{S}$. As the workload of a queue increases excessively, the schedule shifts attention to it and selects available service configurations $S \in \mathcal{S}^e$ that provide more service capacity to that queue, potentially at the expense of others. That lowers the workload at the queue, trading it for increased workload in others and load balancing them.

A strictly diagonal matrix $\mathbf{B}$ induces a direct *simple priority scheme*. That is, as the weight $B_{qq}$ of queue $q$ is increased (while those of others remain constant), the queue attains higher service priority. This results in the queue receiving more service bandwidth over time and enjoying a lower workload.

When $\mathbf{B}$ has negative off-diagonal elements, those have an indirect effect on service priorities, entangling the queues and inducing a *coupled priority scheme*. That is, when $B_{pq} < 0$ with $p \neq q$, the relative priority of queue $p$ decreases as the workload of queue $q$ increases. As $X_q$ grows in size, more attention

needs to be paid in servicing queue $q$, while as $X_p$ grows in size, less attention is paid to queue $q$. It can be seen that the weight $B_{pq} < 0$ induces a specific coupling between the corresponding queues.

We also observe that the proof of stability in section 5 is robust to any sublinear perturbations of information or time. In particular, if there is a switching delay between configurations or an information lag in knowledge of $X(t)$, or some error in the calculation of $\max_S \langle S, \mathbf{B}X \rangle$ then *as long as the corresponding perturbation does not grow linearly with $t$, then rate stability will still be assured*. See [Ross and Bambos, 2009] for more detail on this observation. For example, if any calculation error or delay is bounded, then stability will hold. For clarity in the proofs we have not included additional terms, but the intuition is that any such sublinear term will have no impact on the limiting case as $X(t)$ becomes large.

For some processing systems, there can be computational issues in the requirement to calculate $\max_S \langle S, \mathbf{B}X \rangle$ in real-time over every possible service vector. The geometric structure of the cone schedules helps to overcome this by recognizing that *when the workload vector $X$ is large, any bounded change in workload will move the workload between adjacent cones* (adjacent cones have a common boundary when $\Re^Q$ is divided into the cones $\mathcal{C}_S$). Therefore, cone schedules can be implemented by evaluating $\langle S, \mathbf{B}X \rangle$ over a much smaller subset of service vectors at each point in time. This was shown to be a special case of sublinear perturbations in the cone schedules, and discussed in detail in [Ross and Bambos, 2009].

To conclude, the adjustment of the $Q \times Q$ entries of the matrix $\mathbf{B}$ allows for generating a rich family of stable service schedules. The dynamic priorities of the queues relate directly to the quality of service (QoS) they receive and the workload they see. We are currently exploring such performance issues further.

# 7 Conclusions and Further Research

We have established that the family of Cone Schedules maximizes the system throughput for very general processing systems under very general conditions. These schedules naturally select the best available vectors, and this leads to the maximum possible system throughput. Arrival and service processes are as general as possible, and the stability proofs are presented with minimal assumptions. Previous stability results are generalized here to a setting with generalized service vectors, fluctuating resource availability and continuous time scheduling.

By exploring the analysis from a geometric standpoint we have gleaned important intuition for stability as well as performance and scalability of the schedules. Further research is necessary to deeper explore the effects of changes to the $\mathbf{B}$ matrix. The transition between environments is also a topic for further study, since it may not be trivial, especially as switching costs become important.

# References


[Andrews et al., 2001] Andrews, M., Awerbuch, B., Fernandez, A., Leighton, T., Liu, Z., and Kleinberg, J. (2001). Universal-stability results and performance bounds for greedy contention-resolution protocols. *Journal of the ACM*, 48(1):39–69.

[Anshelevich et al., 2002] Anshelevich, E., Kempe, D., and Kleinberg, J. (2002). Stability of load balancing algorithms in dynamic adversarial systems. In *Proceedings of the thiry-fourth annual ACM symposium on Theory of computing*, STOC '02, pages 399–406, New York, NY, USA. ACM.

[Armony and Bambos, 2003] Armony, M. and Bambos, N. (2003). Queueing dynamics and maximal throughput scheduling in switched processing systems. *Queueing Systems*, 44(3):209.

[Bambos and Michailidis, 2004] Bambos, N. and Michailidis, G. (2004). Queueing and scheduling in random environments. *Advances In Applied Probability*, 36:293–317.

[Bambos and Walrand, 1993] Bambos, N. and Walrand, J. (1993). Scheduling and stability aspects of a general class of parallel processing systems. *Advances in Applied Probability*, 25:176–202.

[Borodin et al., 2001] Borodin, A., Kleinberg, J., Raghavan, P., Sudan, M., and Williamson, D. (2001). Adversarial queuing theory. *Journal of the ACM*, 48(1):13–38.

[Cruz, 1991a] Cruz, R. L. (1991a). A calculus for network delay, part i: Network elements in isolation. *IEEE Transactions on Information Theory*, 37:114–131.

[Cruz, 1991b] Cruz, R. L. (1991b). A calculus for network delay, part ii: Network analysis. *IEEE Transactions on Information Theory*, 37:132–141.

[Dai and Prabhakar, 2000] Dai, J. and Prabhakar, B. (2000). The throughput of data switches with and without speedup. In *INFOCOM 2000. Nineteenth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*, volume 2, pages 556 –564 vol.2.

[Dai and Lin, 2005] Dai, J. G. and Lin, W. (2005). Maximum pressure policies in stochastic processing networks. *Operations Research*, 53(2):197–218.

[Dai and Lin, 2008] Dai, J. G. and Lin, W. (2008). Asymptotitic optimality of maximum pressure policies in stochastic processing networks. *Annals of Applied Probability*, 18(6):2239–2299.

[Hung and Michailidis, 2011] Hung, Y. and Michailidis, G. (2011). Stability and control of acyclic stochastic processing networks with shared resources. *Forthcoming, IEEE Transactions on Automatic Control*.

[Kushner, 2006] Kushner, H. (2006). Control of multi-node mobile communications networks with time-varying channels via stability methods. *Queueing Systems*, 54:317–329.

[Leonardi et al., 2005] Leonardi, E., Mellia, M., Marsan, M. A., and Neri, F. (2005). Joint optimal scheduling and routing for maximum network throughput. In *INFOCOM 2005. 24th Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings IEEE*, volume 4, pages 13–17.

[Marsan et al., 2005] Marsan, M. A., Leonardi, E., Mellia, M., and Neri, F. (2005). On the stability of isolated and interconnected input-queueing switches under multiclass traffic. *IEEE Transactions on Information Theory*, 51(3):1167–1174.

[McKeown et al., 1999] McKeown, N., Mekkittikul, A., Anantharam, V., and Walrand, J. (1999). Achieving 100% throughput in an input-queued switch. *IEEE Transactions on Communications*, 47(8):1260–1267.

[Neely et al., 2003] Neely, M., Modiano, E., and Rohrs, C. (2003). Dynamic power allocation and routing for time varying wireless networks. In *INFOCOM 2003. Twenty-Second Annual Joint Conference of the IEEE Computer and Communications. IEEE Societies*, volume 1, pages 745 – 755 vol.1.

[Ross and Bambos, 2009] Ross, K. and Bambos, N. (2009). Projective cone scheduling (pcs) algorithms for packet switches of maximal throughput. *IEEE Transactions on Networking*, 17(3):976–989.

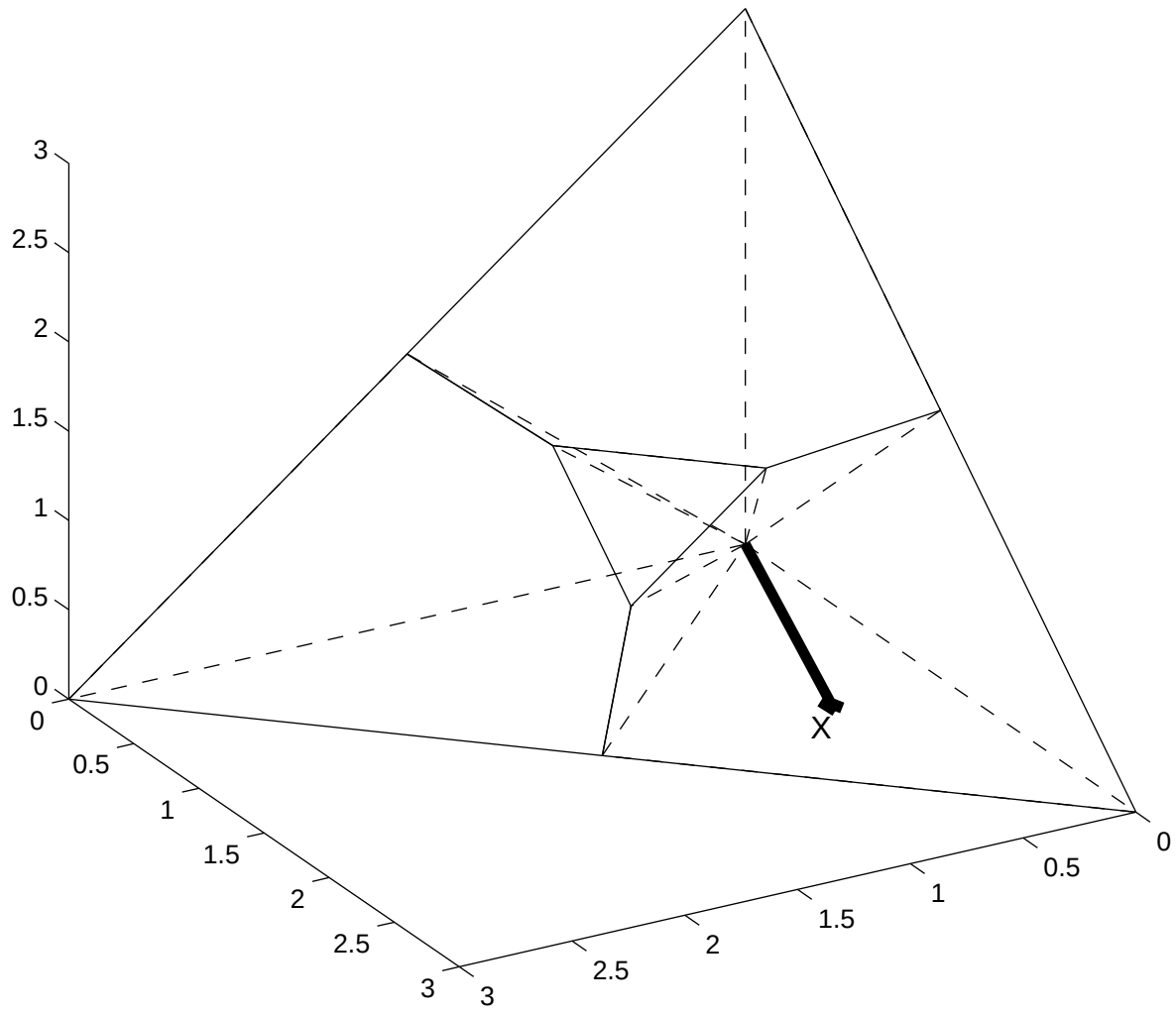
[Stolyar, 2004] Stolyar, A. L. (2004). Maxweight scheduling in a generalized switch: State space collapse and workload minimization in heavy traffic. *The Annals of Applied Probability*, 14(1):pp. 1–53.

[Tassiulas, 1995] Tassiulas, L. (1995). Adaptive back-pressure congestion control based on local information. *IEEE Transactions on Automatic Control*, 40(2):236–250.

[Tassiulas and Bhattacharya, 2000] Tassiulas, L. and Bhattacharya, P. (2000). Allocation of interdependent resources for maximal throughput. *Stochastic Models*, 16(1):27–48.

[Tassiulas and Ephremides, 1992] Tassiulas, L. and Ephremides, A. (1992). Stability properties of constrained queueing systems and scheduling policies for maximum throughput in multihop radio networks. *IEEE Transactions on Automatic Control*, 37(12):1936–1948.

[Tsaparas, 2000] Tsaparas, P. (2000). Stability of open multiclass networks in adversarial queueing theory.


3
2.5
2
1.5
1
0.5
0
0
0.5
1
1.5
2
2.5
3
3
2.5
2
1.5
1
0.5
0
X